



I.P. Sharp Associates

LEVEL

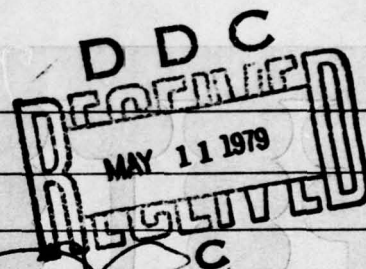
A061406

12

EUCLID COMPILER  
QUARTERLY REPORT No 4

AD A068542

DDC FILE COPY



This document has been approved  
for public release and sale; its  
distribution is unlimited.

70 02 13 008

Quarterly Technical Report  
for  
Euclid Compiler for PDP-11  
Number 4

PERIOD COVERED: 1 October to 31 December 1978

This research was sponsored by  
Defense Advanced Research Projects  
Agency under ARPA Order No. 3475  
Contract No. MDA 903-78-C-0037  
Monitored by Steve Walker  
Effective Date of Contract 1 Oct 1977  
Contract Expiry Date 31 Mar 1979

A portion of this project is being  
sponsored by the Canadian Department  
of National Defence, Chief of Research  
and Development

The views and conclusions in this document are those of the  
author and his associates and should not be interpreted as  
necessarily representing the official policies, either  
expressed or implied, of the Defense Advanced Research  
Projects Agency or the United States Government.

David Bonyun  
I.P. Sharp Associates Limited  
Suite 600  
265 Carling Avenue  
OTTAWA, Canada K1S 2E1

ACCESSION for	
NTIS	Vite Section <input checked="" type="checkbox"/>
DDC	B.R. Section <input type="checkbox"/>
UNANNOUNCED	<input checked="" type="checkbox"/>
JUSTIFICATION	<i>on file</i>
EY	
DISTRIBUTION	CODES
Dist.	SPECIAL
<i>A</i>	

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) <u>6</u> EUCLID Compiler for PDP-11. Report Number 4		5. TYPE OF REPORT & PERIOD COVERED Quarterly Technical to 31 December 1978
7. AUTHOR(s) D.A. Bonyun, I.P. Sharp Associates <u>10</u> David A/Bonyun		6. PERFORMING ORG. REPORT NUMBER <u>14</u> IPSA-3819-0041
9. PERFORMING ORGANIZATION NAME AND ADDRESS I.P. Sharp Associates Limited 600-265 Carling Avenue Ottawa, Ontario, Canada K1S 2E1		8. CONTRACT OR GRANT NUMBER(s) <u>15</u> MDA 903-78-C-0037 X/ARPA Order-3475
11. CONTROLLING OFFICE NAME AND ADDRESS Information Processing Techniques Office DARPA ARLINGTON, VA 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS <u>12</u> I/P
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE <u>11</u> 30 January 1979
		13. NUMBER OF PAGES 9
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) <u>9</u> Quarterly rept. no. 4 1 Oct-31 Dec 78		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Part of this project is funded by the Canadian Department of National Defense		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) EUCLID, compiler, computer security		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → The full Euclid compiler will define a subset of Euclid between Small Euclid (the subset for bootstrapping) and Middle Euclid (the subset requested by the KSOS team). Complexities in this language, as defined, have appeared only during this writing of the compiler and some redefinition of the language seems indicated. ←		

## Quarterly Technical Report #4

### Euclid Compiler Project

#### Report Summary

The Euclid Compiler Project is to provide for the PDP-11/45 computer architecture under the operating system, UNIX, a compiler for the language, Euclid. The compiler is multi-pass, highly portable, and written in subset of Euclid itself. The subset is known as Small Euclid, and was chosen specifically to permit bootstrapping.

The problems associated with this project have been the result of a new, highly sophisticated language requiring, for a relatively small machine, a compiler with features associated with the verifiability of resulting code. Techniques, already in use in compiler writing, had to be adapted to assist in handling the peculiarities of the language and some previously unaddressed problems associated with the production of verifiable code. While the problems are all essentially solved, the time taken to create the compiler has lengthened because of the inherent complexity of the language which was not previously recognized.

The language will probably have to be redefined to some extent because of its inherent complexity. A subset, known as Middle Euclid, requested by the KSOS team, has been defined; and the first compiler, due now in May, will define another subset between Small Euclid and Middle Euclid. Somewhat more effort would permit this compiler to attain Middle Euclid as the language it is capable of handling.

The verification aspects of the language are significant and some of the mechanisms for handling them are in place in various passes of the compiler where they are appropriate. The code to come from the compiler ought to be accompanied by an augmented source stream indicating the insertion of compiler-generated assertions.

Future research is indicated to obtain a stable useful subset of Euclid. The use of this subset, or of the one actually accepted by the compiler in the reworking of KSOS, seems most desirable. The verification of Euclid code also deserves future attention.

## Quarterly Technical Report #4

### Euclid Compiler Project

The Euclid Compiler project was established to create for the PDP-11/45 computer architecture a compiler for the language, Euclid, to work under the operating system, UNIX. It was designed to be a multi-pass compiler, written in Euclid itself (actually a subset specifically chosen to permit bootstrapping and known as Small Euclid), and expected to be as portable to other machines as possible.

The overall process, now nearing completion, consisted of the following steps:

- 1) Define the subset language, Small Euclid.
- 2) Create a transliterator of that subset to the UNIX language, C.
- 3) Adapt the tool, created at the University of Toronto, known as SSL (the Syntax Semantics language), to be available to support the Euclid subset.
- 4) Divide the proposed compiler into its component passes - (6 in all: the Screener/Scanner, the Parser, the Builder, the Conformance Pass, the Allocator, and the Coder) and design each.
- 5) Do the preliminary overall design work including the anticipation of the problems likely to be encountered and some possible solutions. (A library of working papers is available.)
- 6) Write the 6 passes. The first two, complete for some time, were adapted from the original Small Euclid Transliterator. The remaining four are all table driven: the creation of the table is the task of the SSL and the support routines are coded in Small Euclid.
- 7) Test the passes.
- 8) Bootstrap the compiler.

During the execution of this set of tasks several technical problems have arisen. Notably, the interaction of the linguistic features of the language have been noted in great detail and the resulting complexity recognized. As a result the compiler will itself be much larger than anticipated and will handle a language which is not the full Euclid as defined. The limits to the language to be compilable are essentially those imposed by its complexity and by the time available to write the compiler.

The principal conclusion reached during the past few months is that the language itself is much more involved and complex than ever imagined at the beginning of the contract. It is likely that a redefinition will become necessary in the future to permit any enhancements to be built onto a stable and manageable base. The first redefinition, or definition of a subset, will be the de-facto language specified by acceptability to the compiler.

Future research will, in all likelihood, be influenced by this redefinition. Although the language was originally designed to write verifiable code, especially that required for security kernels and other "small" blocks of software, it has become very powerful. The team writing KSOS asked for a subset of Euclid which they believed to be adequate for their tasks (named Middle Euclid) and although it was never made available to them (the work involved being too great and the time needed much more than expected) this definition survives. The first compiler, at the end of April 1979 is likely to be capable of handling more than Small Euclid but less than Middle Euclid.

It is expected that some effort will still be placed in the rewriting of the security-sensitive portion of KSOS in whatever

I

Euclid is available. This is because verification efforts still seem to be oriented towards some of the conditions which Euclid imposes. The Euclid which is available, therefore, will materially affect the way in which these portions of KSOS will be coded.

An effort to permit verification of Euclid code seems to be a desirable future path. As Euclid becomes more widely accepted the amount of code to be verified will surely increase and well established techniques seem to be a requirement in that case.

## Appendix

In keeping with the last quarterly report a table is presented below to indicate for each of the three languages (Small Euclid - the language for bootstrapping the compiler itself; Middle Euclid - the language originally requested by Ford Aerospace for KSOS; Full Euclid - the language described in the report of the Committee) what percentages of the language will be available in the April compiler by each of the 6 passes.

The issues of verifiability are separate. The two aspects of Euclid which enable it to produce verifiable code are:

- (1) Access control (imports, exports, and aliasing);
- (2) Assertions.

It is expected that by April there will be work leading to the easy insertion of features to the compiler to permit (a) its generation of some assertions (by Pass 4 - the Conformance Pass) and (b) its enforcing of imports and exports (by Pass 3, the Builder). Neither feature is likely to be in the April compiler, but both are moderately simple, if time consuming, additions.

Table 1

Percentage of language capabilities to be handled by each pass, April 1979.

<u>Language</u>	<u>Pass 1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
Small Euclid	100	100	100	100	100	100
Middle Euclid	100	100	95	95	95	60
Full Euclid	100	100	95	95	95	50

Distribution List for Technical Reports

ARPA, Attn: Program Management 2 copies  
1400 Wilson Blvd.  
ARLINGTON, VA 22209 U.S.A.

Dr. G.X. Amey 1 copy  
CRAD DST(SE)4  
Department of National Defence  
101 Colonel By Drive  
OTTAWA, Ontario, Canada K1S 0K2

Defense Documentation Center (DDC) 12 copies  
Cameron Station  
ALEXANDRIA, VA 22314 U.S.A.

Letters of Transmittal sent to:

Mr. Dave Keyes  
Science Procurement Branch, CCC  
11C1 Place du Portage, Phase III  
11 Laurier Street  
HULL, Quebec K1A 0S5

Defense Contract Administration Service  
Management Area, Ottawa  
219 Laurier Avenue, West  
6th Floor  
OTTAWA, Canada K1A 0S5

Mr. Steve Walker  
OSD-CCCI  
The Pentagon  
WASHINGTON, D.C. 20310